Generating a Population of Animated faces from Pictures

Won-Sook Lee, Nadia Magnenat-Thalmann
MIRALab, CUI, University of Geneva, Geneva, Switzerland
E-mail: {wslee, thalmann}@cui.unige.ch
Fax: ++ 41 (0)22 705-7780
Tel: ++ 41 (0)22 705-7766
http://www.miralab.unige.ch

Abstract

This paper describes a simple and robust method for generating photo-realistic animated face population in a virtual world. First we make a small set of 3D-virtual faces just using photo data, using a method so called virtual cloning. Then we use very intuitive 3D-morphing system to generate a new population, which is profited from 3D-structure of existing virtual faces. The virtual cloning method uses a set of orthogonal pictures of a person. This efficient method for reconstructing 3D heads suitable for animation starts with the extraction of feature points from the orthogonal picture sets. A previously constructed, animation-ready generic model is transformed to each individualized head based on the features extracted from the orthogonal pictures. Using projections of the 3D head, a 2D texture image is obtained for an individual reconstructed from pictures, which is then fitted to the clone, a fully automated procedure resulting in 360-degree seamless texture mapping. We also introduce an extremely fast dynamic system for 3D morphing with 3D spatial interpolation and powerful 2D texture-image metamorphosis based on triangulation inherited from 3D structure of virtual faces. The interface allows for the real-time inspection and control of the morphing process.

Keywords: Orthogonal photos, deformation, generic model, facial animation, texture mapping, morphing, user interface design.

1. Introduction

Before constructing virtual groups, crowds or populations, we must be able to synthesize realistic figures and move them about in a convincing and efficient way. Animators agree that the most difficult subjects to model and animate realistically are humans and particularly human faces. The explanation resides in the universally shared (with some cultural differences) processes and criteria not only for recognizing people in general, but also for identifying individuals, expressions of emotion and other facial communicative signals, based on the covariation of a large number of partially correlated shape parameters within narrowly constrained ranges. Though some of this may be amusingly conveyed in a schematic manner by various 2D or 3D animation technologies, the results are easily identified as cartoons by the most naive observer, while the complex details and subtle nuances of truly realistic representation remain a daunting challenge for the field.

Once we have a way of modeling specific individuals, how can we make use of a limited number of reconstructed faces in the automatic generation of a large population of distinct individuals of the same general characteristics? In this paper, we describe an efficient and robust method for individualized face modeling, followed by techniques for generating animation-ready populations, based on a number of existing head models. We also introduce a rapid dynamic system, which enables 3D polymorphing with smooth texture variation under intuitive control.

1.1. Background

Approaches to the realistic reconstruction of individuals, some of them with a view to animation, include the use of a laser scanner [10], a stereoscopic camera [4], or an active light stripper [11]. There is increasing interest in utilizing a video stream [15] to reconstruct heads and natural expressions. These methods are not developed enough, however, to be commercialized (such as in a camera-like device) in the near future, in terms of direct input of data for reconstruction and final animation as output. The animation results of Guenter et al.[2] are impressive, but input and output processes are not practical for real-time animation. Pighin et al.[3] get naturalistic results for animation using several views as input. Their techniques have elements in common with our methods, but the aims are different. For example, we are primarily interested here in morphing between a number of different individuals, while their application is to interpolate between different stages of a single facial model. Modeling has also been done from picture data [6][7][8][9][13], detecting features, modifying a given generic model and then mapping texture on it. Not all of these, however, combine sophisticated and reliable shape deformation methods with seamless, high-resolution texture generation and mapping.

Techniques for generating virtual populations have been investigated by DeCarlo et al. [1]. They vary a geometric human face model using random distances between facial points sampled from distributions of anthropometric statistics, but they do not consider a real-life face data as an input and realistic texture variation.

The term morph, short for metamorphosis, has been applied to various computer graphics methods for smoothly transforming geometric models or images. Morphing techniques can be classified into image based methods and object-space methods. In most cases, the image-based methods are used for 2D morphing and the object-space methods for 3D morphing.

Image morphing has three terms to be considered, which are feature specification, warp generation methods, and transition control [22]. These areas relate to the ease of use and quality of results. Feature specification is the most tedious aspect of morphing. Although the choice of allowable primitives may vary, all morphing approaches require careful attention to the precise placement of primitives. Given feature correspondence constraints between both images, a warp function over the whole image plane must be derived. This process, which we refer to as Warp Generation, is essentially an interpolation problem. Another important problem in image morphing is transition control. If transition rates are allowed to vary locally across in-between images, more interesting animations are possible. The main methods are based on mesh warping [23], field morphing [24], radial based functions [25], thin plate splines [26], energy minimization [27], work minimization [28] and multilevel free-form deformations [14]. A tradeoff exists between the complexity of feature specification and warp generation. As feature specification becomes more convenient, warp generation becomes more formidable. Some methods do not guarantee the one-to-one property of the generated warp functions while others derive one-to-one warp functions, but the performance is hampered by its high computational cost. Most of the morphing methods and applications require manual user intervention to specify matches between given objects. In many cases, significant user involvement is required to achieve a pleasing visual result. It is desirable to eliminate, or at least to limit, this involvement in morphing design, and to have a more automatic process. In addition the speed is quite low requiring at least several seconds due to the heavy calculation.

The traditional formulation for image morphing considers only two input images at a time, i.e. the source and target images. Morphing among multiple images, referred as polymorph, is understood to mean a seamless blend of several images at once.

A metamorphosis or a 3D morphing, for example topology independent morphing in a short animation film "Galaxy Sweetheart" [30] in late '80, is the process of continuously transforming one object into another [12][13]. Since there is no intrinsic solution to the morphing problem, user interaction can be a key component of morphing software if two objects do not share the same structure. Most efforts have been dedicated to the correspondence problem, and we still lack intuitive solutions to control the shape interpolation.

The idea of combining 2D-image morphing and 3D-shape morphing has been little exploited. Nevertheless, this seems a natural approach to better results with less effort. Furthermore, were this to be combined with face cloning methods, much user interaction could be avoided.

1.2. Our approach

Our method is focused on easy and intuitive populating of faces in a virtual world. The best way is to make several photo-realistic faces in the same structure and use them to generate new ones out of them in real-time speed. We design a virtual cloning system starting from one generic model, so that every face can share the same structure. Then the next step to generate new population using 3D-morphing system, which use shape and image morphing, but eliminate many computational intensive steps being profited from 3D-face structure.

In Section 2 of this paper, we describe a fast and robust method for head-shape modeling based on the extraction of feature points from two orthogonal views. An individualized head is created by modifying a predefined generic head, using the feature points obtained from two orthogonal views of the individual. A generalized and highly flexible Free Form Deformation is used to deform the shape of the generic model.

The same two orthogonal projections used in feature point extraction also serve as input to a process for texture mapping, as described in Section 3. The texture images and mappings for texture coordinates are generated automatically using geometrical deformation and multiresolution technique. The texture image retains resolution as high as the input images since it does not use the conventional cylindrical projection to combine two input images. These texture images and coordinates are used for image morphing among several given texture images, based on a common triangulation derived from the generic head, to generate new texture mappings for the newly created heads of Section 2. Section 4 is devoted to a dynamic system for generating heads, including shape and texture details that interpolate several input heads. This enables the user to visually control mixing ratios with close to real-time calculation. Some experimental results are presented in Section 5.

2. Head shapes from photos

2.1. Feature detection

2D photos offer cues to the 3D shape of an object. It is not feasible, however, to consider 3D-points densely distributed on the head. In most cases, we know the location of only a few visible features such as eyes, lips and silhouettes, key characteristics for recognizing people. We call them as feature points whose location can be detected either automatically, semi-automatically or at least interactively. To reconstruct a photographically realistic head, ready for animation, we detect corresponding feature points on both of two orthogonal images -- front and side -- and deduce their 3D positions. This information is used to modify a generic model through a geometrical deformation. Our feature detection is processed in a semiautomatic way using the structured snake method [9] with some anchor functionality. Figure 1 depicts an orthogonal pair of normalized images, showing the features detected. Here normalization signifies locating images in the feature point space, so that the front and side views of a head have the same height. The feature points are overlaid on the images even though they are located in spaces with different origins and scales from the images.

The two 2D sets of position coordinates, from front and side views, i.e., the *(x, y)* and the *(z, y)* planes, are combined to give a single set of 3D points. To get perfectly aligned and orthogonal views is almost impossible, and this leads to difficulties in determining the *(x, y, z)* coordinates of a point from the *(x, y)* on the front image and the *(y, z)* on the side image. Taking the average of the two points often results in an unnatural face shape. Thus we rely mainly on the front *y* coordinate, using the side *y* only when we do not have front one. This convention is very effective when applied to not_perfectly_orthogonal pairs of images. In addition, for asymmetrical faces, this convention allows for retention of the asymmetry with regard to the most salient features, even though a single side image is used in reconstructing both the right and left aspects of the face.

A global transformation moves the 3D feature points to the space containing the generic head, which is a normalization of every head.

2.2. Modification of a generic model

We have a certain set of 3D feature points. The problem is how to deform a generic model, which has more than a thousand points to make an individualized smooth surface. One solution is to use 3D feature points as a set of control points for a deformation. Then the deformation of a surface can be seen as an interpolation of the displacements of the control points.

Free-Form Deformations (FFD) [16] belong to a wider class of geometric deformation tools. However FFD has a serious constraint in that control points boxes have to be rectangular, which limits the expression of any point of the surface to deform relative to the control points box. Farin [19] extends the natural neighbors' interpolant based on the natural neighbors' coordinates, the Sibson coordinate system [17] based on Voronoi/Dirichlet and Delaunay diagrams [18] for a scattered data interpolant, using the support for a multivariate Bezier simplex [20]. He defines a new type of surfaces with this extended interpolant called Dirichlet surfaces. Combining FFD's and Dirichlet surfaces leads to a generalized model of FFD's: Dirichlet FFD's or DFFD's.

In the Dirichlet-based FFD approach, any point of the surface to deform located in the convex hull of a set of control points in general position, is expressed relative to a subset of the control points set with the Sibson coordinate system. One major advantage of this technique is that it removes any constraint on the position and topology of control points. It also removes the need to specify the control lattice topology [21]. One control point is defined at the position of each surface point, so that any displacement applied to the control point will also be applied to the surface point.

Therefore DFFD is used here to get new geometrical coordinates for a modification of the generic head on which are situated the newly detected feature points. All points on this head are located utilizing the feature points as constraint control points for DFFD. Since our feature points includes the front and side silhouette, the convex hull of control points contains most points on a 3D head, but there can be some missing points outside the convex hull. So we add 27 extra points surrounding the 3D head in Figure 1. The correct shapes of the eyes and teeth are assured through translation and scaling appropriate to new head. As shown in Figure 1, this is a rough matching method; it does not attempt to locate all points on the head exactly, in contrast to range data from for example laser scanners, which create enormous numbers of 3D-points. However considering the input data (pictures from only two views), the result is quite respectable. Most important, it greatly limits the size of the data set associated with reasonable shape of an individual head, as it is necessary to accelerate animation speed.

To improve realism, we make use of automatic texture mapping, as described in the next section.

Figure 1: Modification of a generic head according to feature points detected on pictures. Points on a 3D head are control points for DFFD.

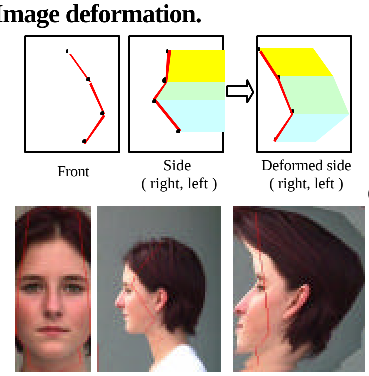
3. Seamless texture mapping

Texture mapping serves not only to disguise the roughness of shape matching determined by a few feature point, but also to imbue the face with more realistic complexion and tint. If the texture mapping is not correct, the accurate shape is useless in practice. We use information from the set of feature points detected to generate texture fully automatically, based on the two views. We then obtain appropriate texture coordinates for every point on the head using the same image transformation.

3.1. Texture generation

The main criterion is to obtain the highest resolution possible for most detailed portions. We first connect two pictures along predefined feature lines using geometrical deformations and, to avoid visible boundary effects, a multiresolution technique.

3.1.1. Image deformation.



(a)

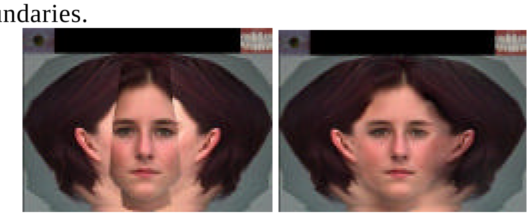
(b)

Figure 2: (a) Red lines are feature lines. (b) Original two photos and a deformed side photo.

We use the front view preponderantly, since it provides the highest resolution for the features. The side view is deformed to be connected to front view along certain defined feature points lines on the left hand side and, flipped over, on the right hand side. The feature lines are decided automatically on the front image as visualized in Figure 2 by the red lines. A corresponding feature line is also automatically defined for the side images. We deform the side image to transform the feature line to coincide with the one on the front view. Image pixels on the right side of the feature line are transformed by the same transform as the line transform. To get the right part of the image, we utilize the side image as it is and deform it according to the right-hand red feature line on the front image. For a left part of the image, we flip the side image and deform it according to the left-hand red feature line on the front image. The resulting three images are illustrated in Figure 3 (a). Here we use a piecewise linear transformation using piecewise feature lines, but higher degree feature curves can perform a smoother deformation for side images.

3.1.2. Multiresolution image mosaic

The three images resulting from the deformation are merged using a pyramid decomposition [5] based on the Gaussian operator. We utilize REDUCE (to reduce the size of the image) and EXPAND (to expand the size of the image) operators to obtain G_k (the Gaussian image) and L_k (the Laplacian image) for a step k. Then we connect three L_k images at each level on any given curve; here they are the feature lines described above. Then the connected image P_k is augmented to obtain S_k, which is the combination of P_k and S_{k+1}. The final image is S_0. This multiresolution technique is effective in removing the boundaries between the three images. No matter how carefully the picture-taking environment is controlled, in practice boundaries are always visible. As in Figure 3 (a), skin colors are not continuous when the multiresolution technique is not applied. The image in (b) shows the results with the technique, which has smoothed the connection between the images without visible boundaries.



(a) (b)

Figure 3: Combining the texture images generated from the three (front, right and left) images without multiresolution techniques, in (a) and with the technique in (b).

Eye and teeth images are superimposed automatically on the top of the image as shown in Figure 3. These are important for animation of eye and lips.

3.2. Texture coordinate fitting

To find suitable coordinates on a combined image for every point on a head, we first project an individualized 3D head onto three planes as shown in left side of Figure 4. Guided by the feature lines used for image merging in above section, we decide to which plane a point on a 3D head is to be projected. The points projected on one of three planes are then transferred to either the front or the side 2D-feature point space. Finally, we transform them to the space of texture image to obtain the final texture coordinates and the final mapping of points on a texture image is generated. The origins of each space are also shown in Figure 4. The 3D head space is the space for the 3D head model, the 2D feature point space is the one for feature points used for feature detection, the 2D image space is the one for the orthogonal images used for input, and the 2D texture image space is for the generated image space. The 2D-feature point space is different from the 2D-image space even though they are displayed together in Figure 1.

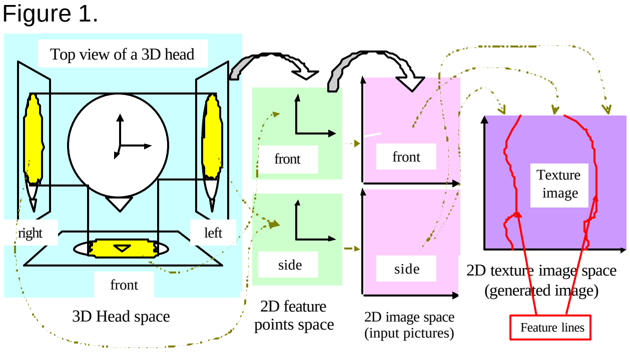


Figure 4: **texture coordinate fitting.**

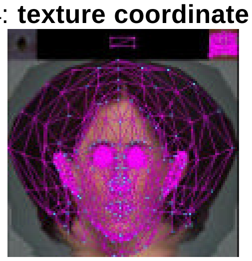


Figure 5: **texture coordinates on an image.**

The final texture fitting on a texture image is shown in Figure 5. The final texture mapping results in smoothly connected images inside triangles of texture coordinate points, which are accurately positioned. The eye and teeth fitting process are done with predefined coordinates and transformations related to the texture image size. After doing eyes and teeth mapping once for a generic model, it is fully automated. The triangles are a projection of triangular faces on a 3D head. Since our generic model is endowed with a triangular mesh, the texture mapping benefits from an efficient triangulation of texture image containing finer triangles over the highly curved and/or highly articulated regions of the face and larger triangles elsewhere, as in the generic model. This triangulation is also used for the 3D-image morphing in Section 4.

3.3. Cloning Results

Figure 6 shows several views of the head reconstructed from the two pictures in Figure 1.

Figure 6: **Snapshots of a reconstructed head in several views.**

Other examples covering wide range of age and many kinds of ethnic group are shown in left side of Figure 11. Every face in this paper is modified from the SAME generic model shown in Figure 1.

3.3.1. Shape-texture separation

Good pairs of orthogonal photos are hard to come by, unless we do the photography ourselves in a controlled situation. Sometimes sets of images are available, some of which have clearer shape while others have good resolution. The front view in Figure 8 is satisfactory both for shape and for texture resolution while the middle side view has better resolution but the rightmost side view has outline clear. For best results, we can combine shape data from the latter with texture image date from the former. The process is outlined in Figure 7.

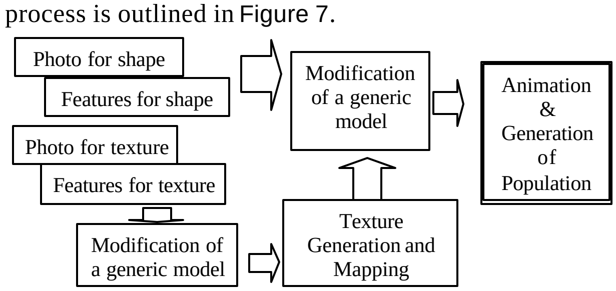


Figure 7: A procedure for reconstruction from shape-image-separated input photos.

3.3.2. Facial animation

Figure 8: **An example of a reconstructed head from three photos and several expressions on it.**

The predefined regions [29] of the generic model are associated with animation information, which can be directly transported to the heads constructed from it by geometrical modification in Section 2. Figure 8 shows several expressions on a head reconstructed from three pictures.

4. 3D morphing System

Morphing technology has seen a great deal of development, either in 2D or in 3D. In this paper, we vary aspects of both 2D and 3D representations in creating new virtual faces, showing how easy and fast the smooth morphing can be achieved. When we morph one person to another in 3D, we must deal with alteration in both shape and texture. Since every face created through the methods described here shares the same topology, it is relatively easy to vary head shapes in 3D just using a simple linear interpolation of 3D coordinates. On the other hand, it is less straightforward to carry out 2D morphing of textures since this requires some feature information specifying correspondences between specific points in the two images. Here we use a surprisingly simple method, drawing on 3D information to facilitate 2D morphing among texture images.

4.1. Texture morphing

To create a new texture image and coordinates, we use texture polymorph with a given set of cloned heads with textures. Image morphing normally considers feature specification, warp generation, and transition control. When we, however, use texture image with 3D head information, feature specification and warp generations are automatically given by 3D head topology and only a simple calculation based on triangulation with given ratios among input texture images is performed.

Two steps are necessary to obtain texture morphing. First, texture coordinates are interpolated, then image morphing is carried out. We use simple linear interpolation among several texture coordinate sets.

4.1.1. Image metamorphosis based on triangulation

We morph several images with given ratios using the information of texture coordinates and the triangulation information of the texture mapping. We first interpolate every 3D vertex on the several heads. Then to generate a new intermediate texture image, we morph triangle by triangle the entire 3D head. The parts of the image used for texture mapping are triangulated by projection of triangular faces of 3D heads since the generic head is a triangular mesh as seen in Figure 5. With this information for triangles, barycentric coordinate interpolation can be employed for image morphing. First, the three vertexes of each triangle are interpolated. Then pixel values inside triangles can be obtained from interpolation between corresponding pixels in several triangles with the same barycentric coordinate. Smoothing of the image pixels is achieved through bilinear interpolation among four neighboring pixels. Given two texture images, for instance, we find, the barycentric coordinate (u, v, w) for each pixel (x, y) in an intermediate images inside a triangle $P_1 P_2 P_3$. The corresponding triangles in the two input images are $P_{1L} P_{2L} P_{3L}$ and $P_{1R} P_{2R} P_{3R}$, respectively. Then $uP_{1L} + vP_{2L} + wP_{3L}$ is the corresponding pixel in the first image and $uP_{1R} + vP_{2R} + wP_{3R}$ in the second. The color value $M(x, y)$ of a given pixel (x, y) is found from linear interpolation, with given ratios r_i where $\sum_{i=1}^{n} r_i = 1$ and $n=2$, between the color value $M_L(x, y)$ of the first image and $M_R(x, y)$ of the second as in the formula below.

The resulting head manifests very smooth images without any gaps in the textured parts. Figure 9 shows the final result of 3D morphing. It depicts the results of interpolating the shapes, the skin and the hair colors between an older Caucasian male and a younger Asian female.

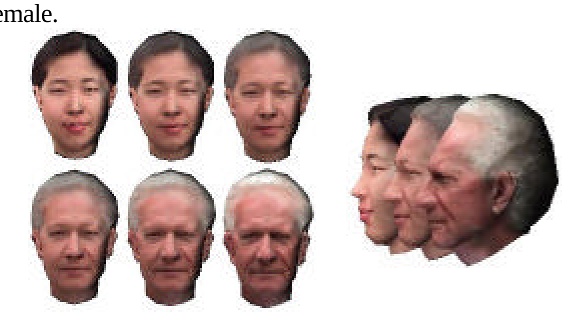


Figure 9: **Morphing steps between two.**

This approach has the advantage of extending to several people in a natural way. Figure 10 illustrates a dynamic system for 3D morphing among a given number of people. The interface has two windows, the left one for controlling input ratio and the upper right one containing the result. Two options are provided, one-time morphing and continuous morphing. One-time morphing provides a convenient way to select an input ratio in an n-polygon in the left window for n given virtual persons. The result appears immediately. For continuous morphing, the result varies according to the ratio points, which are chosen along a line or curve in the polygon. Since only barycentric calculation for texture image pixels is required, the calculation is very fast. There is also an option to set on/off for shape and image variation separately.

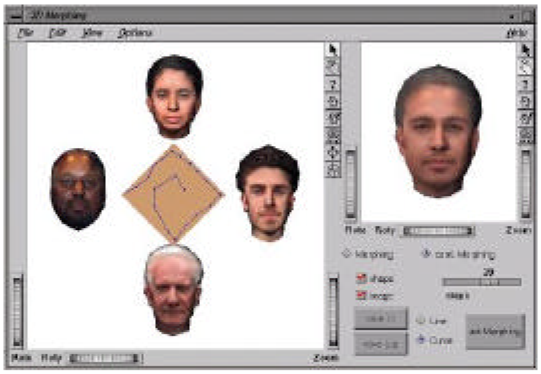
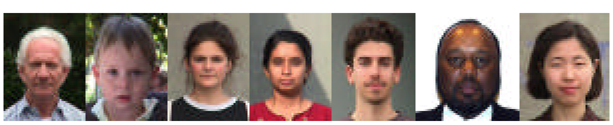


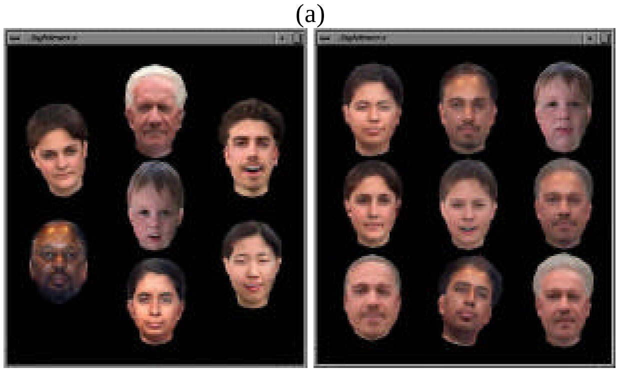
Figure 10**: A dynamic system for 3D morphing. Here four persons are used for continuous morphing following a user-specified curve.**

5. Results

We report here on experiments with cloning and with creating new heads from clones. We used seven orthogonal photo sets as inputs. Figure 11 (a) shows the input photos while (b) shows output. Included are four Caucasians of various ages, an Indian, an Asian, and an African. There are three adult females, three adult males and a child. The creation of a population is also illustrated on the right side on bottom in Figure 11 (c). The heads in (c) do not have correspondences in the real world while ones in (b) have corresponding person in the real world. Figure 11 (b) and (c) are snapshots of an *OpenGL*-based interface allowing the animation of a few heads.



(a)



(b) (c)

Figure 11: (a) 7 photos of real-people. (b) a snapshot of animation of 7 virtual clones. (c) a snapshot with created 9 heads from heads in (b).

All faces are animation-ready. The size of the texture image for each person is 256x256, which has less than 10 KB. The total amount of data for the heads in *OpenInventor* format is small considering their realistic appearance. The size of Inventor format (corresponding to *VRML* format) is about 200 KB. The texture image is stored in *JPEG* format and has sized about 5~50 KB depending on the quality of pictures; all examples shown in this paper have size less than 45 KB. The number of points on a head is 1257, where 192 of them are for teeth and 282 of them are used for the two eyes. This leaves only 783 points for the individualization of the face surface aside from eyes and teeth. The small size of data used in the face geometry and texture gives us the possibility of animating several heads together in real time. Every head has its own location and script file, which defines several continuous expressions (like eye, eyebrow, mouth movement or head rotation) with a given range of time. The expressions repeat with different frequency for each head. With smaller size of texture images and fewer points on heads, we will certainly increase number of heads for real-time animation.

6. Conclusion

We have introduced a suite of methods for the generation of large populations of photo-realistic faces, enabled for immediate real-time animation, from just a few pairs of orthogonal pictures. For the easy and fast generation of new virtual human, we first virtual-clone some people using only one generic model. Then we populate big number of virtual human based on them under an intuitive and rapid control.

One key to our technique is the efficient reconstruction of animation-ready individualized faces fitted with fully automatic seamless textures. This technique was robust enough to allow one operator to clone some 70 individuals in five days in Orbit'98, Basel, Switzerland. It was also demonstrated in CeBit'99, Hannover, Germany. The procedure is universal, applicable to men and women, adults and children, and different races, all using the same generic model.

To generate a population from a small number of heads is based on the reconstruction technique. As we intend, all models generated in this system have the same topological structure, i.e. same triangulation, for the 3D shape and similar characteristics for the texture images, enabling real-time simulation of 3D metamorphosis among several virtual heads. Control of the mixing ratio for shape and image is intuitive and the rotation in any view allows the full appreciation of the 3D virtual heads. Various facial expressions can also be dynamically mixed in this system.

In the experiments for this paper, variation of hairstyle was limited, restricting the apparent diversity of the population. Ongoing researche is individualized hair and body reconstruction from pictorial input.

7. Acknowledgment

The authors would like to thank every member of MIRALab, especially Laurent Moccozet, Pierre Beylot, and Hyewon Seo. We are also grateful to Chris Joslin for proof reading this document. This project is supported by Swiss National Research Foundation and a European project eRENA.

8. References

[1] Douglas DeCarlo, Dimitris Metaxas and Matthew Stone, "An Anthropometric Face Model using Variational Techniques", In *Computer Graphics (Proc. SIGGRAPH'98)*, pp. 67-74, 1998.

[2] Brian Guenter, Cindy Grimm, Daniel Wood, "Making Faces", In *Computer Graphics (Proc. SIGGRAPH'98)*, pp. 55-66, 1998.

[3] Frederic Pighin, Jamie Hecker, Dani Lischinski, Richard Szeliski, David H. Salesin, Synthesizing "Realistic Facial Expressions from Photographs", In *Computer Graphics (Proc. SIGGRAPH'98)*, pp. 75-84, 1998.

[4] http://www.turing.gla.ac.uk/turing/copyrigh.htm

[5] Peter J. Burt and Edward H. Andelson, "A Multiresolution Spline with Application to Image Mosaics", *ACM Transactions on Graphics*, 2(4):217-236, Oct., 1983.

[6] Tsuneya Kurihara and Kiyoshi Arai, "A Transformation Method for Modeling and Animation of the Human Face from Photographs", In *Proc. Computer Animation'91*, Springer-Verlag Tokyo, pp. 45-58, 1991.

[7] Takaaki Akimoto, Yasuhito Suenaga, and Richard S. Wallace, Automatic Creation of 3D Facial Models, IEEE Computer Graphics & Applications, Sep., 1993

[8] Horace H.S. Ip, Lijin Yin, Constructing a 3D individual head model from two orthogonal views. The Visual Computer, Springer-Verlag, 12:254-266, 1996.

[9] Lee W. S., Kalra P., Magnenat-Thalmann N, "Model Based Face Reconstruction for Animation", In *Proc. Multimedia Modeling (MMM'97)*, Singapore, pp. 323-338, 1997.

[10] Yuencheng Lee, Demetri Terzopoulos, and Keith Waters, "Realistic Modeling for Facial Animation", In *Computer Graphics (Proc. SIGGRAPH'96)*, pp. 55-62, 1996.

[11] Marc Proesmans, Luc Van Gool. "Reading between the lines - a method for extracting dynamic 3D with texture". In *Proc. of VRST'97*, pp. 95-102, 1997.

[12] J. Kent, W. Carlson , R. Parent, "Shape Transformation for Polygon Objects", In *Computer Graphics (Proc. SIGGRAPH'92)*, pp. 47-54, 1992.

[13] Kiyoshi Arai, Tsuneya Kurihara, Ken-ichi Anjyo, "Bilinear interpolation for facial expression and metamorphosis in real-time animation", *Visual Computer*, Vol. 12, No. 3, 1996.

[14] S.-Y. Lee, K.-Y. Chwa, S.-Y. Shin, G. Wolberg, "Image metamorphosis using Snakes and Free-Form deformations", In *Computer Graphics (Proc. SIGGRAPH'95)*, pp. 439-448, 1995.

[15] P. Fua, "Face Models from Uncalibrated Video Sequences", In *Proc. CAPTECH'98*, pp. 215-228, 1998.

[16] Sederberg T. W., Parry S. R., "Free-Form Deformation of Solid Geometric Models", In *Computer Graphics (Proc. SIGGRAPH'86)*, pp. 151-160, 1986.

[17] Sibson R., "A Vector Identity for the Dirichlet Tessellation", *Math. Proc. Cambridge Philos. Soc.*, 87, pp. 151-155, 1980.

[18] Aurenhammer F., "Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure", *ACM Computing Survey*, 23, 3, September 1991.

[19] Farin G., "Surface Over Dirichlet Tessellations", *Computer Aided Geometric Design*, 7, pp. 281-292, North-Holland, 1990.

[20] DeRose T.D., "Composing Bezier Simplexes", *ACM Transactions on Graphics*, 7(3), pp. 198-221, 1988.

[21] Moccozet L., Magnenat Thalmann N., "Dirichlet Free-Form Deformations and their Application to Hand Simulation", In *Proc. Computer Animation'97*, IEEE Computer Society, pp.93-102, 1997.

[22] George Wolberg, "Image morphing: a survey", *Visual Computer*, International Journal of Computer Graphics, Vol. 14, No. 8/9, pp.360-372, 1998.

[23] Wolberg G. "Digital image warping", IEEE Computer Society Press, Los Anamitos, Calif, 1990.

[24] T. Beier, S. Neely, "Feature-based image metamorphosis", In *Computer Graphics (Proc. SIGGRAPH'92)*, pp. 35-42, 1992.

[25] Arad N., Dyn N, Reisfeld D, Yeshurun Y. "Image warping by radial basis functions : applications to facial expressions", *CVGIP:Graph Models Image Processing*, 56:161-172, 1994.

[26] Litwinowicz P., Williams L., "Animating images with drawings", In *Computer Graphics (Proc. SIGGRAPH'94)*, pp. 409-412, 1994.

[27] Lee S., Chaw K-Y, Hahn J., Shin SY, "Image morphing using deformation techniques", *Journal Visualization Computer Animation*, 7:3-23, 1996.

[28] Gao P., Sederberg TW, "A work minimization approach to image morphing", *Visual Computer*, International Journal of Computer Graphics, Vol. 14, No. 8/9, pp. 390-400, 1998.

[29] Kalra P, Mangili A, Magnenat-Thalmann N, Thalmann D, "Simulation of Facial Muscle Actions Based on Rational

Free Form Deformations", *Proc. Eurographics'92*, pp. 59-69, NCC Blackwell,1992.

[30] N. Magnenat-Thalmann, M.De Angelis, T. Hong, D. Thalmann, "Design Transformation and Animation of Human Faces", *Visual Computer*, Springer, Vol. 5, No. 1-2, pp. 32-39, 1989.